10301493   SUPPLIER NUMBER: 20869896   (THIS IS THE FULL TEXT)
Routing orderpickers in a warehouse: a comparison between optimal and
  heuristic solutions.
Koster, Rene de; Poort, Edo van der
IIE Transactions, v30, n5, p469(12)
May, 1998
ISSN: 0740-817X   LANGUAGE: English   RECORD TYPE: Fulltext; Abstract
WORD COUNT: 7831   LINE COUNT: 00615

AUTHOR ABSTRACT: In this paper the problem of finding efficient
orderpicking routes is studied for both conventional warehouses, where
pickers have a central depot for picking up and depositing carts and pick
lists, and modern warehouses, where orderpicking trucks can pick up and
deposit pallets at the head of every aisle without returning to the depot.
Such environments can be found in many warehouses where paperless picking
is performed from pallet locations with pickers having mobile terminals
receiving instructions one by one. In order to find orderpicking routes
with a minimal length in both the situations of a central depot or
decentralized depositing, we extend the well-known polynomial algorithm of
Ratliff and Rosenthal (1) that considered warehouses with a central depot.
In practice, the problem is mainly solved by using the so-called S-shape
heuristic in which orderpickers move in a S-shape curve along the pick
locations. The performance of the new algorithm and the S-shape heuristic
are compared in three realistic orderpicking systems: (1) narrow-aisle
high-bay pallet warehouse; (2) picking in shelf area with decentralized
depositing of picked items; and (3) conventional orderpicking from
wide-aisle pallet locations. The new algorithm gives a reduction in travel
time per route of between 7 and 34%. It turns out that the reduction in
travel time strongly depends on the lay-out and operation of the warehouse.

TEXT:
    1. Orderpicking in warehouses
    In warehouses and distribution centers, products have to be picked
from specified storage locations on the basis of customer orders. In
general, the orderpicking process is the most laborious of all warehouse
processes. It may consume as much as 60% of all labor activities in the
warehouse (2). Especially in distribution environments, the pick process is
usually carried out under time constraints. Orders tend, more and more, to
arrive late and have to be shipped the same day at pre-fixed departure
times per area of destination. This leads to peak loadings and an on-going
pressure to carry out the orderpicking process as efficiently as possible.
Therefore, many warehouses nowadays use paperless orderpicking systems

rather than picklists containing the picking locations that have to be collected at a central printer. The most commonly used way of paperless orderpicking is via mobile, handheld or vehicle-mounted, terminals and printers. Paperless orderpicking systems have the clear advantage that orderpickers and storers are connected on-line with the warehouse information system, which results in accurate up-to-date stock information, on-line reaction to exceptional situations, and on-line control of progress. Moreover, the orderpickers can obtain, pick and store instructions without leaving the storage area. These aspects lead to pick-error reduction and increased productivity. The savings may be substantial in view of the picking throughput time per destination, but also in view of the efficient use of expensive special orderpicking equipment such as high-bay narrow-aisle orderpicking trucks. The use of mobile terminals offers the possibility of a more decentralized way of operation. For example, in warehouses where orderpicking trucks are used and empty pick carriers are available at the head of all aisles, orderpicking trucks can drop off full pallets at the head of each aisle. The transportation of the full pallets is taken care of by faster and also cheaper equipment, such as conveyors and forklift trucks. The orderpickers may therefore finish a picking route in any aisle and proceed with the new route in the same aisle. In the sequel of the paper such systems will be called orderpicking systems with decentralized depositing.

Another way to achieve savings on orderpickers and equipment is by optimizing orderpicking routes. Given that the orderpicker has to collect a number of products in specified quantities at known locations, in what sequence should the orderpicker visit these locations in order to minimize the distance traveled? The problem of finding shortest orderpicking routes for warehouses with a central depot can be solved in running time as being linear in the number of aisles and the number of pick locations (1, 3). In van Dal (4) the algorithm is extended for different warehouse lay-outs. More recently, Gelders and Heeremans (5) solved the orderpicking problem by applying the branch-and-bound algorithm of Little et al. (6) to a simplified warehouse lay-out for a particular type of warehouse. They report reductions on the total walking distance in the warehouse varying between 9 and 40% (depending on the number of items to be picked). The problem of finding a shortest orderpicking route in the case of decentralized depositing has not been considered in the literature. In practice, the problem of finding orderpicking routes in a warehouse is mainly solved by the so-called S-shape heuristic in which orderpickers move in a S-shape curve along the pick locations skipping the aisles where nothing has to be picked. More advanced heuristics are considered in Hall (7).

In practice, the optimal routing algorithms are infrequently used. This is partly due to the fact that the algorithm is not commonly known. Also the application of the algorithm to lay-outs different to the model containing parallel aisles and a central depot has not been considered in

the literature. In order to use the algorithm, it has to be incorporated into the existing warehouse management system software, which means a change in the core functionality of such a system. Furthermore, the savings produced by using optimal algorithms in practice are not clear a priori. We also have to keep in mind that there are extra expenses and risks associated with implementing the optimal algorithm, because the algorithm is complex (based on dynamic programming) and not transparent. Finally, note that the routing of orderpickers is not in all cases the process where the largest gain in efficiency may be obtained. Substantial gains can also be obtained from proper clustering the orders into routes (8-10).

In this paper, we will show that the polynomial algorithm of Ratliff and Rosenthal (1) can be extended in such a way that shortest orderpicking routes can be found in both warehouses with a central depot and warehouses with decentralized depositing. Moreover, we will investigate the real gain in travel and total route time of the optimal algorithm in comparison with the S-Shape heuristic. based on practice, we consider the following three warehousing situations:

1. Picking with orderpicking trucks in a narrow-aisle high-bay pallet warehouse;

2. Manual picking from shelf racks with decentralized depositing on, for example, a conveyor ;

3. Picking in a conventional wide-aisle pallet warehouse using an orderpicking truck without lifting capability.

The paper is organized as follows: A mathematical model for the orderpicking problem is considered in Section 2.

Section 3 presents the extension of the algorithm of Ratliff and Rosenthal (1). The numerical results of comparing the performance of the new algorithm and the S-shape heuristic in the three practical orderpicking systems are discussed in Section 4.

2. The orderpicking problem

A warehouse consists of a number of aisles of equal length with the items being stored at both sides of the aisles. Trucks use the aisles to pick up items. They can traverse the aisles in both directions, and changing direction is not a problem. Each order consists of a number of items that are usually spread out over a number of aisles. We assume that the items of an order can be picked in a single route. Aisle changes are possible at the front and rear ends of the aisles. Aisle changes with an orderpicking truck or crane to neighboring aisles are often very time consuming. We assume that the trucks are capable of simultaneously driving and lifting. In a warehouse with a central depot the orderpick carts and pick lists can be picked up and deposited at the depot. In this case the start and finish point of the orderpicking route are known beforehand. In paperless orderpicking systems with decentralized depositing the orderpick carts and pick lists can be picked up and deposited at the head of every aisle. In this case only the start point of the orderpicking route is known beforehand. In order to determine an orderpicking route of minimum length

the travel time between each pair of locations in the warehouse needs to be specified. In the specification of the travel time we can take into account the time for entering an aisle and the time for accelerating and decelerating while driving from one location to another. Note that we will only minimize the travel time as the other orderpicking activities have to be performed anyway and therefore do not impact the choice of an orderpicking route. The warehouse lay-out is shown schematically in Fig. 1.

We will model the warehouse with the orderpicking locations as an undirected (multi) graph with the vertices corresponding to the pick locations and endpoints of the aisles, the edges indicating whether two locations or endpoints in the warehouse are connected directly, and the length of the edges indicating the travel times in the warehouse. A depot vertex is added together with edges connecting the depot vertex with the head of the aisles. More formally, let N denote the number of aisles in the warehouse and M the number of items that have to be picked in the orderpicking route. We define an undirected (multi) graph G with the vertex set V and the edge set E with vertices $(v.sub.i)$ representing the location of item i, for i = 1, ..., M, vertices $(a.sub.j)$ and $(b.sub.j)$ representing the ends of aisle j, for j = 1, ..., N, and vertex s denoting the depot. For each pair of vertices u, v in V corresponding to adjacent locations in the warehouse, the edges {u, v}, {u,v} are introduced (the edge is mentioned twice to stress that two edges are added to the multi-graph), and the length of each of these edges, denoted by d(u, v), is defined as the travel time between the corresponding locations in the warehouse. In a warehouse with a central depot at the head of an aisle, say aisle p, the edges {s, $(b.sub.p)$}, {s, $(b.sub.p)$} are introduced. In a warehouse with decentralized depositing starting from the head of an aisle, say aisle p, the edge {s, $(b.sub.p)$} is introduced together with the edges {s, $(b.sub.j)$} for j = 1, ..., N. Figure 2 shows the graph for the warehouse and pick locations of Fig. 1 in the case of decentralized depositing starting from $(b.sub.4)$.

Any orderpicking route will be considered as being a special kind of subgraph of the warehouse graph G, and is therefore called a routesubgraph. More formally, any subgraph T of G is called a routesubgraph if the edges form a cycle in G that includes s and each of the vertices $(v.sub.i)$ at least once. The length of subgraph T is defined as the sum of the length of the edges in T. Figure 3 shows a routesubgraph for the warehouse lay-out of Fig. 1 in the case of decentralized depositing starting from the head of aisle 4.

In Ratliff and Rosenthal (1) an algorithm is presented that constructs an orderpicking route from a given routesubgraph. The problem of finding a shortest orderpicking route can therefore be solved by finding a routesubgraph of minimum length. In the Appendix necessary and sufficient conditions are given for a graph being a routesubgraph (see Theorem A2).

3. A polynomial algorithm for finding a minimum length routesubgraph

In this section we extend the optimal algorithm of Ratliff and

Rosenthal (1) in such a way that shortest orderpicking routes can be found in a warehouse with either a central depot or with decentralized depositing. We wish to stress that a shortest orderpicking route in a warehouse with decentralized depositing can not be simply obtained from a shortest orderpicking route in a warehouse with a central depot by leaving out some of the edges. Figure 4 shows the shortest orderpicking routes in the warehouse lay-out of Fig. 1 for both the situations of a central depot at the head of aisle 4 and that of decentralized depositing starting from the head of aisle 4. The shortest orderpicking routes have a length of 94 and 88, respectively. Note that leaving out the connection between aisles 4 and 5 in the shortest orderpicking route in the warehouse with a central depot at the head of aisle 4 gives an orderpicking route of length of 92 for the warehouse with decentralized depositing.

3.1. Partial routesubgraphs

Every subgraph of G using only the edges of aisle 1 that can be extended into a routesubgraph is called a (L.sub.1)-partial routesubgraph. Any (L.sub.1)-partial routesubgraph can be extended with the vertices and edges of aisle 2 in order to obtain a (L.sub.2)-partial routesubgraph. If we continue in this way, we finally get the (L.sub.N)-partial routesubgraphs which are precisely the routesubgraphs. More formally, define (L.sub.j), for j = 1, . . ., n, as the subgraph of G that contains the vertices and edges of the first j aisles of the warehouse (that is: including all double edges {(a.sub.i-1), (a.sub.i)} and double edges {(b.sub.i - 1), (b.sub.i)} for i (less than or equal to) j). We distinguish the following three subgraphs of (L.sub.j):

a. The subgraph (Mathematical Expression Omitted) that includes (a.sub.j), (b.sub.j) and the edges {(a.sub.j-1), (a.sub.j)}, {(a.sub.j-1), (a.sub.j)}, {(b.sub.j-1), (b.sub.j)}, {(b.sub.j-1), (b.sub.j)} as well as the vertices and edges of the first j-1 aisles (that is: including the edges {(b.sub.i-1), s}, for i (less than or equal to) j-1, of which one edge is double in case p (less than or equal to) j - 1);

b. The extension of the (Mathematical Expression Omitted) subgraph, denoted by subgraph (Mathematical Expression Omitted), with the vertex s and the edges between s and (b.sub.j);

c. The extension of the (Mathematical Expression Omitted) subgraph, denoted by (Mathematical Expression Omitted), with the vertices and edges of aisle j connecting the vertices between (a.sub.j) and (b.sub.j).

Let L be any subgraph of G. The graph G\L is the subgraph of G that is obtained by deleting all edges in L. Any subgraph T of L is called a L-partial routesubgraph if there exists a subgraph C of G\L such that T (union) C is a routesubgraph of G. C is called a completion of T. Any L-partial routesubgraphs (T.sub.1) and (T.sub.2) are called equivalent if each completion C in G\L of (T.sub.1) is also a completion of (T.sub.2), and vice versa. For any vertex v in G, the parity of v is 0 if the degree of v is 0, odd if the degree of v is odd, and even otherwise. Any subgraph T of G is called a component of G if for all vertices v, w in T there is a

path in T from v to w and there is no vertex in G\T with a path to a vertex in T. In order to find all equivalent (L_j)-partial routesubgraphs, we can use the following result.

Theorem 1. Any (L_j)-partial routesubgraphs (T_1) and (T_2) are equivalent if and only if

1. (a_j) and (b_j) have the same parity in (T_1) and (T_2);

2. (T_1) and (T_2) have the same number of components; and

3. (T_1) and (T_2) are connected to s by the same number of edges.

Proof. The proof is similar to the proof of Theorem 3 in Ratliff and Rosenthal (1). Condition (3) is added.

As a consequence of Theorem 1, we can represent the equivalence classes of (L_j)-partial routesubgraphs by the quadruple (parity of (a_j), parity of (b_j), number of components, number of edges connecting s). The parity of (a_j) and (b_j) can be 0, e (even) or u (uneven), the number of components can be 0, 1, or 2, and the number of edges connecting s can be 0, 1, or 2. The number of components of an (L_j)partial routesubgraph can never be more than 2, since the completion of the graph can only connect via the aisle ends to two components at maximum. In principle there exist 3 x 3 x 3 x 3 = 81 possible quadruples. However, it will be shown in the Appendix that only the following equivalence classes correspond with (L_j)-partial routesubgraphs.

Theorem 2. The following quadruples are the only equivalence classes of (L_j)-partial routesubgraphs:

(0, e, 1, 0), (e, 0, 1, 0), (u, u, 1, 0), (e, e, 1, 0), (e, e, 2, 0),

(0, 0, 0, 0), (0, u, 1, 1), (u, 0, 1, 1), (u, e, 1, 1),

(u, e, 2, 1), (e, u, 1, 1), (e, u, 2, 1), (0, e, 1, 2), (e, 0, 1, 2),

(u, u, 1, 2), (e, e, 1, 2), (e, e, 2, 2).

Now recall that each routesubgraph contains the edge $\{s, (b_p)\}$. Consequently, the (Mathematical Expression Omitted)-partial routesubgraphs in the equivalence classes (0, e, 1, 0), (e, 0, 1, 0), (u, u, 1, 0), (e, e, 1, 0), (e, e, 2, 0) only exist for j smaller than p and the (Mathematical Expression Omitted)-partial routesubgraphs in the equivalence classes (0, e, 1, 2), (e, 0, 1, 2), (u, u, 1, 2), (e, e, 1, 2), (e, e, 2, 2) only exist for j at least equal to p. Hence, we can denote the equivalence classes, without loss of generality, as the triples:

(0, 0, 0), (0, e, 1), (e, 0, 1), (u, u, 1), (e, e, 1), (e, e, 2), (0, u, 1), (u, 0, 1), (u, e, 1), (u, e, 2), (e, u, 1), (e, u, 2).

Note that in comparison with Ratliff and Rosenthal (1) the equivalence classes (0, u, 1), (u, 0, 1), (u, e, 1), (u, e, 2), (e, u, 1), (e, u, 2) have been added and the equivalence class (0, 0, 1) has been left out. This class is deleted, because we start in the first and finish in the last aisle containing items. Note that (0, 0, 0) is the only equivalence class of (Mathematical Expression Omitted)-partial routesubgraphs. Finally,

observe that $(e, 0, 1)$, $(0, e, 1)$, and $(e, e, 1)$ are the only equivalence classes of (Mathematical Expression Omitted)-partial routesubgraphs, so that a routesubgraph of minimal length can be found from the (Mathematical Expression Omitted)-partial routesubgraphs in the equivalence classes $(e,0,1)$, $(0,e,1)$, and $(e,e, 1)$ (1).

3.2. Finding minimum length $(L.sub.N)$-partial routesubgraphs

In each of the equivalence classes a smallest $(L.sub.N)$-partial routesubgraph can be found by applying dynamic programming (11). This algorithm starts in aisle 1 and considers the vertices and edges of the following aisle in each successive step of the algorithm. In order to use dynamic programming, we have to define the potential states, the possible transitions between states, and the costs involved in such a transition. The states correspond to the equivalence classes of $(L.sub.j)$-partial routesubgraphs. In order to determine the transitions between states when adding the vertices and edges of a new aisle, we consider the transition from aisle j-1 to aisle j in three steps: First from (Mathematical Expression Omitted) to (Mathematical Expression Omitted), next from (Mathematical Expression Omitted) to (Mathematical Expression Omitted), and finally from (Mathematical Expression Omitted) to (Mathematical Expression Omitted). All transitions can be found by applying Theorem A1 in the Appendix. The cost of each transition is equal to the sum of the lengths of the edges added in the transition. For each transition, it holds that in each equivalence class a shortest partial routesubgraph can be found by taking the smallest of the partial routesubgraphs that after the transition end up in this equivalence class. We now consider the three transitions in more detail.

3.2.1. The transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted)

This transition determines how to add the edges $\{(a.sub.j-1),(a.sub.j)\}$, $\{(a.sub.j-1),(a.sub.j)\}$, $\{(b.sub.j-1),(b.sub.j)\}$, $\{(b.sub.j-1),(b.sub.j)\}$ to the partial routesubgraphs in the $((L.sub.i-1).sup.+)$ equivalence classes in order to obtain the partial routesubgraphs in each of the (Mathematical Expression Omitted) equivalence classes. The eight possible ways to traverse the corresponding edges are shown in Fig. 5.

Note that the transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted) does not change the number of edges connecting the depot vertex. (TABULAR DATA FOR TABLE 1 OMITTED) The equivalence classes that we obtain by combining the eight ways with the partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are given in Table 1. For example, applying possibility (1) to the (Mathematical Expression Omitted)-partial routesubgraph in the class $(u,e,1)$ gives a (Mathematical Expression Omitted)-partial routesubgraph in the class $(u,0,1)$. For each (Mathematical Expression Omitted) equivalence class a minimum length partial routesubgraph is found by taking the smallest of all partial routesubgraphs that end up in this equivalence

class. For example, a smallest (Mathematical Expression Omitted)-partial routesubgraph in the class (u,0,1) can be obtained by taking the smallest of the (Mathematical Expression Omitted)-partial routesubgraphs in the classes (u,0,1) and (u,e,1), and adding the edge {(a.sub.j-l),(a.sub.j)}.

3.2.2. The transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted)

This transition can change the number of edges connecting the depot vertex. Note that the transition depends on the number of edges between s and (b.sub.j) in the warehouse graph. If there is no edge between s and (b.sub.j) then the (Mathematical Expression Omitted)partial routesubgraphs in each of the equivalence classes are already the (Mathematical Expression Omitted)-partial routesubgraphs. Now consider the case in which there is one edge between s and (b.sub.j). In this case we have to choose whether or not to traverse the edge {s,(b.sub.j)}. These possibilities are labeled (1) and (2) in Fig. 6. The equivalence classes that are obtained by combining the two ways with the partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are given in Table 2.

Finally, consider the case that there are two edges between s and (b.sub.j). Now there are three possibilities to traverse the edges between s and (b.sub.j), see Fig. 6. The equivalence classes that we obtain by combining the three ways with the partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are given in Table 3. Note that possibility (1) is not possible, because at least one edge {s,(b.sub.j)} is included in every routesubgraph.

(TABULAR DATA FOR TABLE 3 OMITTED)

In Table 3 there are two exceptional transitions. Consider first the transition from (u,e,2) to (u,u,1), denoted by (1), by adding the edge {s,(b.sub.j)}. Any (Mathematical Expression Omitted)-partial routesubgraph in the equivalence class (u,e,2) has two components, one containing (a.sub.j) and s, and one containing (b.sub.j) (see Lemma A6 in the Appendix). Adding the edge {s,(b.sub.j)} therefore reduces the number of components by one. Consider now the transition from (e,u,2) to (e,e,2) denoted by (2). In this case any (Mathematical Expression Omitted)-partial routesubgraph has two components, one containing (a.sub.j) and one containing s and (b.sub.j) (see Lemma A6). Therefore adding the edge {s,(b.sub.j)} does not reduce the number of components.

3.2.3. The transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted)

This transition determines how to traverse the edges connecting the items in aisle j. Again, the depot connection remains unchanged. It appears that there are only six ways to traverse the edges in aisle j, see Fig. 7.

In possibility (5) only the longest double edge in aisle j is not traversed. Again, the cost of each possibility is equal to the sum of the corresponding edge lengths. Note that transitions (3) and (4) may be applied only if the aisle contains at least one item, transition (5) may be applied only if the aisle contains at least two items, and transition (6)

may be applied only if the aisle contains no items.

The equivalence classes that we obtain by combining the six ways with the partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are given in Table 4. For example, applying possibility (3) to the (Mathematical Expression Omitted)-partial routesubgraph in the class (0,e,1) gives a (Mathematical Expression Omitted)-partial routesubgraph in the class (e,e,2). This concludes the description of the algorithm for finding a minimum routesubgraph. The algorithm considers all aisles and items, and for each aisle and item a constant number of operations has to be done. Hence, the time-complexity of the algorithm is O(max(N,M)).

3.3. Example

We conclude this section by applying the algorithm to the warehouse and orderpicking locations of Fig. 1 for the case of decentralized depositing starting from the head of aisle 4. Recall that the corresponding warehouse graph is shown in Fig. 3. In Table 5 the length of the smallest (TABULAR DATA FOR TABLE 4 OMITTED) partial routesubgraph is given for all equivalence classes and states. The equivalence classes in Table 5 are numbered. Each entry in the table has the form l(c,n), where l is the length of the smallest partial routesubgraph in this equivalence class that is obtained by extending the partial routesubgraph in the equivalence class c by traversing the new edges according to the transition with number n (in Roman numbers). Each entry that does not correspond to a partial routesubgraph, or a partial routesubgraph that can never have a minimum length, is denoted by (-). For example, any partial routesubgraph that does not start in aisle 4 can never yield a routesubgraph with a minimum length.

The table is constructed by working from left to right. Recall that (0,0,0) is the only (Mathematical Expression Omitted) equivalence class. The partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are found by applying Table 2, since there is one edge between s and (b.sub.1). From the (Mathematical Expression Omitted) equivalence class (0,0,0), we obtain (Mathematical Expression Omitted)-partial routesubgraphs in the class (0,0,0), by not adding {s,(b.sub.j)}, and in (0,u,1) by adding {s,(b.sub.j)}. The partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are found by applying Table 4. For example, traversing the edges in aisle 1 according to transition number (IV), at a cost of 22, gives a minimum length (Mathematical Expression Omitted)-partial routesubgraph in the equivalence class (0,e,1). The partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are found by applying Table 1. For instance, the partial routesubgraph in the (Mathematical Expression Omitted) equivalence class (0,e,1) of length 26 is found by adding the edges {(a.sub.1),(a.sub.2)} and {(a.sub.1),(a.sub.2)} to the partial routesubgraph in the (Mathematical Expression Omitted) equivalence class (0,e,1). The minimum length partial routesubgraphs in the (Mathematical Expression Omitted) equivalence classes are found by repeatedly applying

the transition tables.

The last column of Table 5 shows that a smallest routesubgraph can be found in the (Mathematical Expression Omitted) equivalence class (e,e,1) with a length of 88. Table 5 can also be used to construct the routesubgraph by working backwards from right to left (see the shaded boxes). The (Mathematical Expression Omitted) partial routesubgraph in the class (e,e,1) was found from the (Mathematical Expression Omitted) (TABULAR DATA FOR TABLE 5 OMITTED) partial routesubgraph in the class (u,u,1) with transition (I), which was found from the (Mathematical Expression Omitted) partial routesubgraph in the class (u,u,1) with transition (I), which again was found from the (Mathematical Expression Omitted) partial routesubgraph in the class (u,u,1) with transition (V), et cetera. The minimum length routesubgraph that can be obtained in this way is shown in Fig. 3 and the resulting orderpicking route is shown in Fig. 4. Note that the orderpicking route ends at the head of aisle 1.

4. A numerical comparison between optimal and heuristic solutions

This section compares the optimal and heuristic solutions for three practical orderpicking systems; namely: (1) narrow-aisle high-bay pallet warehouse; (2) picking in shelf area with decentralized depositing of picked items; and (3) conventional orderpicking from wide-aisle pallet locations. In all configurations, we consider the results of 1000 runs each consisting of a fixed number of randomly distributed picking locations over the aisles in the warehouse. For each experiment, the travel and total time are calculated for both the S-shape heuristic and the optimal algorithm. The resulting average travel (walking or driving) and total route time are given for both algorithms. The total route time includes not only the travel time but also the time for other activities such as picking and remaining tasks including the dropping off and acquisition of pick carriers, administrative tasks and the like. The S-shape heuristic is adapted for the case of decentralized depositing by starting the S-shape curve in the left or right-most aisle where items have to be picked, that is closest to the starting point, and the endpoint of the current route is the starting point of the next route. We wish to mention here that in warehouses with decentralized depositing, there is in fact an additional opportunity for savings on travel and total time when a set of orders has to be picked on an order by order basis by finding the best sequence of orders. This problem will not be considered in this paper, and in all experiments the endpoint of the current route is taken as the starting point of the next route.

4.1. Narrow-aisle high-bay pallet warehouse

This kind of orderpicking system is often used for the picking of fairly large, slowly moving, items that are (TABULAR DATA FOR TABLE 6 OMITTED) stored in pallet racks. The aisles are 50 meters long, and each orderpicking truck has a picking area that consists of either 3 or 4 aisles with 5 or 10 pick locations. The average travel speed within the aisles is 1.5 meters per second and outside the aisles 1 meter per second. The

distance between two neighboring aisles is 4.3 meters. The time needed to leave or enter an aisle is 15 seconds, and the time needed for picking operations and remaining activities is 150 seconds per item. As can be seen in Table 6 the difference in performance between the optimal algorithm and the S-shape heuristic is much smaller when the ratio of stops per aisle is high.

4.2. Picking in shelf area with decentralized depositing of picked items

We now consider a shelf store area where orderpickers pick items in batch with a small pick cart. They receive pick instructions via a mobile terminal with a label printer. At the front end of the aisles, there is a conveyor where the picked items can be dropped off. After dropping off the picked items, the pickers receive information about the items to be picked in the next route. The aisles have a length of 10 meters and each orderpicker has to do 7, 10, or 15 aisles with 20 locations per route. The average walking speed within and outside the aisles is 0.6 meters (TABULAR DATA FOR TABLE 7 OMITTED) (TABULAR DATA FOR TABLE 8 OMITTED) per second.

The distance between two neighboring aisles is 2.4 meters. No additional time is needed for aisle changing. The time needed for the picking operations and remaining activities is 140 seconds per item. Table 7 shows that the savings in walking time obtained by using the optimal algorithm can be substantial. Again the savings depend on the ratio of the number of stops per aisle. However, in this case the relative difference in total time per route is much smaller.

4.3. Conventional orderpicking

Finally, we consider the orderpicking in wide aisles containing pallet racks where the picking is done from the lowest levels. No special orderpicking trucks with lifting capability are necessary. Picking is done with small orderpicking trucks, in special devices, such as roll cages. This situation can be encountered in large retailer warehouses where the picking is done per customer order, the customer being a single store or a number of stores. We assume that the picking is done on the left and right sides of the aisles in a single move, and we neglect the aisle cross-over time, although the aisles in such a warehouse are relatively wide. In such an environment, the roll cages often have to be picked up and deposited, after picking, at a central point. The optimal and heuristic solutions are compared for a warehouse with an aisle length of 40 meters, 8 or 10 aisles per truck, and 10 or 15 stops per route. The central depot is located at the front end of aisle 5. The average travel speed is 0.8 meters per second within and outside the aisles. No additional time is needed to leave or enter an aisle. The distance between two neighboring aisles is 5.5 meters, and the time needed for the picking operations and remaining activities is 210 seconds per item. Table 8 shows that for this type of situation considerable gains can be obtained by using the optimal algorithm instead of the S-shape heuristic.

5. Concluding remarks

The numerical results suggest that the savings in travel time may be substantial when using the optimal algorithm instead of the S-shape heuristic. It appears (for example from Table 7) that the proportional gain in average travel and total time strongly depends on the average number of items per aisle. This can be intuitively argued by realizing that if the density of items per aisle increases then the likelihood also increases that with the optimal algorithm the whole aisle has to be traversed. This does not necessarily mean, that if the ratio of items over aisles increases, the relative performance of the S-shape heuristic increases. Table 6 shows that the case of 10 items and 3 aisles (ratio 3.33) has a performance gap in average driving time of 10.0% and the case of 4 aisles and 10 items (ratio 2.5) has a performance gap in average driving time of 7.3%. For a fixed number of aisles however, an increasing number of items means a better relative performance of the S-shape heuristic.

There are also other factors that influence the proportional gain in travel and total time, for instance the time needed to enter an aisle. If this time increases, then it will become unattractive to enter aisles more than once, and consequently the difference in travel time between the optimal and heuristic solutions will decrease. The times to enter aisles used in the simulation experiments are chosen realistically for the case of orderpicking trucks. If aisle-changing cranes are used instead of trucks then the time for entering aisles will be substantially higher.

The difference in performance of the optimal and heuristic algorithms in high-bay warehouses using orderpicking trucks is limited, especially when the total time is considered and as opposed to solely the travel time. In practical situations the travel time is often about 50% of the total time. In many modern warehouses, the narrow-aisle high-bay orderpicking trucks that have to travel more than one aisle are replaced by multi-functional trucks, i.e., the so-called combi-trucks. These combitrucks are not only responsible for orderpicking, but also perform different tasks such as the storage and retrieval of full pallets. The main advantage of using such trucks instead of conventional orderpicking trucks lies of course in their flexibility. For these multi-purpose trucks, the potential gain of optimal algorithms is limited. Nevertheless, there are many orderpicking systems where the optimal algorithm can lead to substantial savings in total route time and hence in the number of pickers and orderpicking devices.

For a real-life implementation, not only the potential time benefits are important, but also the risks and costs involved. One such risk is the calculation time. Computer calculation times may play a role, if real-time performance becomes important. This may be the case when an error situation occurs: the item demanded is not available at the expected location and a new route has to be calculated on the spot and displayed on the mobile terminal. In many environments however, real-time performance is not important. Calculation times of the optimal algorithm do not play an

important role in the investigated situations. Although the calculation time of the optimal algorithm is up to 15 times longer for the investigated cases than that of the S-shape heuristic, it still only takes milliseconds for a single instance (on a Pentium 120 computer).

References

(1) Ratliff, H.D. and Rosenthal, A.S. (1983) Orderpicking in a rectangular warehouse: a solvable case of the traveling salesman problem. Operations Research, 31, 507-521.

(2) Drury, J. (1988) Towards more efficient orderpicking, in IMM Monographs 1, Institute of Materials Management, Cranfield, UK.

(3) Carlier, J. and Villon, P. (1987) A well-solved case of the traveling salesman problem. Technical Report, University of Compiegne.

(4) van Dal, I.C. (1992) Special cases of the traveling salesman problem. Ph.D. thesis, University of Groningen, Groningen, The Netherlands.

(5) Gelders, L. and Heeremans, D. (1994) Het traveling salesman probleem toegepast op orderpicking. Tijdschrift voor Economie en Management, 19, 381-388.

(6) Little, J.D.C., Murty, K.G., Sweeney, D.W. and Karel, C. (1963) An, algorithm for the traveling salesman problem. Operations Research, 11, 972-989.

(7) Hall, R.W.H. (1993) Distance approximations for routing manual pickers in a warehouse. IIE Transactions, 25, 76-87.

(8) Gibson, D.R. and Sharp, G.P. (1992) Order batching procedures. European Journal of Operational Research, 58, 57-67.

(9) Rosenwein, M.B. (1996) A comparison of heuristics for the problem of batching orders for warehouse selection. International Journal of Production Research, 34(3), 657-664.

(10) de Koster, M.B.M., van der Poort, E.S. and Wolters, M. (1997) Efficient orderbatching methods in warehouses. Working paper, Rotterdam School of Management, Erasmus University Rotterdam.

(11) Bertsekas, D.P. (1976) Dynamic Programming and Stochastic Control, Academic Press, New York.

Appendix

In this appendix we will prove Theorem 2. We use the following results. The following theorem gives necessary and sufficient conditions for any subgraph of G being a (L_j)-partial routesubgraph.

Theorem A1 (Ratliff and Rosenthal (1))

Let j (element of) $\{1, ..., N-1\}$. Any T (subset or equal to) (L_j) is a (L_j)-partial routesubgraph if and only if

1. for each (v_i) (element of) (L_j), the degree of (v_i) is positive in T;

2. each vertex in $T\backslash\{(a_j), (b_j), s\}$, has even degree; and

3. apart from vertices with zero degree, T exists of (a) zero

components, (b) one component with either (a.sub.j), (b.sub.j), or (a.sub.j) and hi contained in this component, or (c) two components with (a.sub.j) in the one and (b.sub.j) in the other component.

The following theorem gives necessary and sufficient conditions for any subgraph of G being a routesubgraph.

Theorem A2 (Ratliff and Rosenthal (1))

Any T (subset or equal to) G is a routesubgraph if and only if:

1. for each (v.sub.i) (element of) V, the degree of (v.sub.j) is positive in T;

2. for each v (element of) T, the degree is even; and

3. apart from vertices with zero degree, T consists of one component.

As a direct consequence of Theorem A2, we obtain that only (e,0,1), (0,e,1), and (e,e,1) are equivalence classes of (Mathematical Expression Omitted)-partial routesubgraphs, and consequently a routesubgraph of minimal length can be found from the (Mathematical Expression Omitted)-partial routesubgraphs in these equivalence classes. As a corollary of Theorem A1, Ratliff and Rosenthal proved the following result.

Corollary A3 (Ratliff and Rosenthal (1))

For each partitioning (P,(P.sup.*)) of the vertices in a routesubgraph, the number of edges between P and (P.sup.*) is even.

Applying Corollary A3 to the warehouse graph of Section 2 gives the following results.

Lemma A4 For j = 1, ..., N,

1. let ((a.sub.j).sup.*) and ((b.sub.j).sup.*) be the adjacent vertices in aisle j of (a.sub.j) and (b.sub.j), respectively. If aisle j is non-empty then every (Mathematical Expression Omitted) partial routesubgraph contains an even number of edges in {(a.sub.j),((a.sub.j).sup.*)}, {(a.sub.j),((a.sub.j).sup.*)}, {(b.sub.j),((b.sub.j).sup.*)}, {(b.sub.j),((b.sub.j).sup.*)};

2. every (Mathematical Expression Omitted)-partial routesubgraph (j (greater than) 1) with zero or two edges connecting s contains an even number of edges in {(a.sub.j-1),(a.sub.j)}, {(a.sub.j-1),(a.sub.j)}, {(b.sub.j-1),(b.sub.j)}, {(b.sub.j-1),(b.sub.j)};

3. every (Mathematical Expression Omitted)-partial routesubgraph (j (greater than) 1) with one edge connecting s contains an odd number of edges in {(a.sub.j-1),(a.sub.j)}, {(a.sub.j-1),(a.sub.j)}, {(b.sub.j-1),(b.sub.j)}, {(b.sub.j-1),(b.sub.j)}.

Proof. (1) Let (Mathematical Expression Omitted) be any (Mathematical Expression Omitted)-partial routesubgraph, and let T denote the corresponding routesubgraph. Define P as the set of vertices corresponding to the items in aisle j. Then, Corollary A3 implies that T, and hence also (Mathematical Expression Omitted), contains an even number of edges in {(a.sub.j), ((a.sub.j).sup.*)}, {(a.sub.j),((a.sub.j).sup.*)}, {(b.sub.j),((b.sub.j).sup.*)}, {(b.sub.j),((b.sub.j).sup.*)}.

(2) Let (Mathematical Expression Omitted) be any (Mathematical Expression Omitted)-partial routesubgraph, and let T denote the

corresponding routesubgraph. We first consider the case that there are no edges connecting s. Define P as the set of vertices in (Mathematical Expression Omitted). Since there are no edges between P and s, Corollary A3 implies that T, and hence also (Mathematical Expression Omitted), contains an even number of edges in {(a.sub.j-1),(a.sub.j)}, {(a.sub.j-1),(a.sub.j)}, {(b.sub.j-1),(b.sub.j)}, {(b.sub.j-1),(b.sub.j)}. Consider now the case that there are two edges connecting s. Define P as the set of vertices in (Mathematical Expression Omitted) together with s. Since there are no edges between (P.sup.*) and s, Corollary A3 implies that T, and hence also (Mathematical Expression Omitted), contains an even number of edges in {(a.sub.j-1),(a.sub.j)}, {(a.sub.j-1), (a.sub.j)}, {(b.sub.j-1),(b.sub.j)}, {(b.sub.j)-1,(b.sub.j)}

(3) Let (Mathematical Expression Omitted) be any (Mathematical Expression Omitted)-partial routesubgraph, and let T denote the corresponding routesubgraph. Define P as the set of vertices in (Mathematical Expression Omitted) together with s. Theorem A2 implies that there is one edge from (P.sup.*) to s. Hence, Corollary A3 implies that T, and therefore (Mathematical Expression Omitted), contains an odd number of edges in {(a.sub.j-1),(a.sub.j)}, {(a.sub.j-1),(a.sub.j)}, {(b.sub.j-1),(b.sub.j)}, {(b.sub.j-1),(b.sub.j)}.

Lemma A4 will now be used to prove an even stronger result that reduces the number of equivalence classes drastically.

Lemma A5 For j = 1, ..., N - 1,

1. in each (Mathematical Expression Omitted), (Mathematical Expression Omitted), and (Mathematical Expression Omitted)-partial routesubgraph with either zero or two edges connecting s, the degree of (a.sub.j) plus the degree of (b.sub.j) is zero or even.

2. in each (Mathematical Expression Omitted) (j (greater than) 1), (Mathematical Expression Omitted), and (Mathematical Expression Omitted)-partial routesubgraph with one edge connecting s, the degree of (a.sub.j) plus the degree of (b.sub.j) is odd.

Proof. (1) Suppose, to the contrary, that there exists a (Mathematical Expression Omitted)-partial routesubgraph with zero or two edges connecting s for which the degree of (a.sub.j) plus the degree of (b.sub.j) is odd. From Lemma A4, we know that after the transitions from (Mathematical Expression Omitted) to (Mathematical Expression Omitted) and from (Mathematical Expression Omitted) to (Mathematical Expression Omitted) the degree of (a.sub.j) plus the degree of (b.sub.j) is still odd when the depot connection has not changed. Them Lemma A4 implies that in the transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted) an even number of edges in {(a.sub.j),(a.sub.j+1)}, {(a.sub.j),(a.sub.j+1)}, {(b.sub.j),(b.sub.j+1)}, {(b.sub.j),(b.sub.j+1)} is added to the (Mathematical Expression Omitted)-partial routesubgraph. But, then either the degree of (a.sub.j) or the degree of (b.sub.j) is odd in (Mathematical Expression Omitted), which is a contradiction with Theorem A1. This proves that in each (Mathematical Expression Omitted),

(Mathematical Expression Omitted), and (Mathematical Expression Omitted)-partial routesubgraph with either zero or two edges connecting s, the degree of (a.sub.j) plus the degree of (b.sub.j) is zero or even.

(2) Can be proven in a similar way and is therefore omitted here.

Combining Theorem A1 and Lemma A5 gives the equivalence classes of (L.sub.j)-partial routesubgraphs in Theorem 2.

We conclude the appendix with a lemma that is useful in the transition from (Mathematical Expression Omitted) to (Mathematical Expression Omitted).

Lemma A6 Let T be a (Mathematical Expression Omitted)-partial routesubgraph, for j = 1, ..., N, with one edge connecting s consisting of two components. If degree (a.sub.j) is odd and degree (b.sub.j) is even, then (a.sub.j) and s are contained in one component and (b.sub.j) in the other, and vice versa.

Proof. Let T be such a partial routesubgraph and let C denote the completion of T. Assume that the degree of (a.sub.j) is odd and the degree of (b.sub.j) is even. Suppose, to the contrary, that (a.sub.j) is contained in component (T.sub.1) and s,(b.sub.j) in component (T.sub.2). Consider the partitioning (P,(P.sup.*)) of the vertices with P defined as the set of vertices in the union of C and (T.sub.2). From Corollary A3, we know that there is an even number of edges between P and (P.sup.*). P can only be connected to (P.sup.*) via (a.sub.j). Recall that (a.sub.j) has odd degree in (T.sub.1). Hence, (a.sub.j) has also odd degree in T (union) C which leads to a contradiction by using Theorem A2. Hence, (a.sub.j) and s are contained in one component and (b.sub.j) in the other. The case that the degree of (a.sub.j) is even and the degree of (b.sub.j) is odd can be proven in the same way.

Biographies

Rene de Koster is a Professor in Logistics/Operations Management at the Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands. He has been involved in the design and realization of several distribution centres and manufacturing plants. His research interests include (control of) material handling systems, warehousing and manufacturing systems.

Edo van der Poort has just finished his Ph.D.-thesis on sensitivity analysis for the Traveling Salesman Problem at the University of Groningen in the Netherlands. His research interests include combinatorial optimization, warehousing, and mathematical modeling.